TINE Release 4.0 News

(April 18, 2008: That was the week that was !)

TINE Kernel 4.0.1

- Version Extension ++
- Primary Additions
  - CF_IMAGE + DIMAGE type

```
typedef struct
{
  TImageSourceHeader sourceHeader;
  TImageFrameHeader frameHeader;
  UINT32 frameBufferSize;
  BYTE *frameBuffer;
} DIMAGE;
```

    - as for CF_SPECTRUM -> data type is a header + array of some type
    - has a header and a body (+ body format)
    - data payload address is assigned in the data type (NEW!)
    - Usage implies (client side):
      - Link queue = 0
      - Suppress duplication = FALSE (no double buffering for comparisons)
  - SetPropertyBuffer(eqmName, prpName, buffer)
    - Fixes an array buffer of the developer's choosing to the given property (no double/triple buffering)
    - No Preset Memory prior to call to equipment function (property handler)

TINE Kernel 4.0.1

- Primary Additions
  - API extensions to allow a reference in all dispatch routine
  - E.g. Equipment Module Registration:

```
TINE_EXPORT int RegisterEquipmentModule(
    char *expName,
    char *eqm,
    int ndevices,
    int (*fcn)(char *,char *,DTYPE *,DTYPE *,short),
    void (*ini)(void),
    void (*tsk)(void),
    int rate,
    void (*exi)(void)
);
```

'Normal' registration routine

```
TINE_EXPORT int RegisterEquipmentModuleEx(
    char *expName,
    char *eqmName,
    int numdevices,
    int (*fcn)(char *,char *,DTYPE *,DTYPE *,short,void *),
    void (*ini)(void *),
    void (*tsk)(void *),
    int rate,
    void (*exi)(void *),
    void *reference
);
```

'Extended' registration routine

Pass the reference registered

TINE Kernel 4.0.1

- Primary Additions
  - API extensions to allow a reference in all dispatch routine
  - E.g. Asynchronous Links

```
TINE_EXPORT int AttachLinkEx2(
    const char *devName,
    const char *devProperty,
    DTYPE *dout,
    DTYPE *din,
    short access,
    int pollingRate,
    void (*callback)(int,int,void *),
    int mode,
    UINT32 callbackID,
    void *reference);
```

Wow! AttachLink has been extended twice !

Callback is fired with the originally passed reference

TINE Kernel 4.0.1

- Secondary Additions
  - API break:
    - AssignDeviceListToProperty()
      - now takes NAME64 list as argument
      - was NAME16
      - Only user was CDI Lib ?
  - Local communication on Windows using shared memory:
    - allowed for console applications using tine32.dll
    - now uses a message loop on its own thread
      - avoids latency of waiting for socket activity (select())
      - (LabView hiccup repaired)
  - HISTORY_HOME will now point to a parallel directory to FEC_HOME called “HISTORY” if not set.
    - e.g. FEC_HOME points to L:\servers\bpm\bin
    - => history files land in L:\servers\bpm\HISTORY (if HISTORY_HOME not set)

TINE Kernel 4.0.1

- Secondary Additions
  - More complete stock alarm definitions
    - e.g. some alarm relevant TINE system error codes (e.g. ‘sedac_error’, or ‘hardware_error’, etc.)
    - handle use of system error codes in a minimal, generic fashion.
    - **Developer can always ‘override’ the system alarm definitions**

TINE Kernel 4.0.1

```
typedef struct ADStag    /* Alarm Definition Structure */
{
   char alarmTag[ALARM_TAG_SIZE];
   UINT32 alarmCode;  /* alarm code ID */
   UINT32 alarmMask;   /* alarm mask */
   UINT16 alarmSystem;/* alarm system ID */
   short  alarmSeverity;/* alarm severity */
   BYTE alarmDataFormat;
   BYTE alarmDataArraySize;
   BYTE reserved[2];
   char alarmText[ALARM_TEXT_SIZE];
   char deviceText[ALARM_TEXT_SIZE];
   char dataText[ALARM_TEXT_SIZE];
   char url[ALARM_TEXT_LONGSIZE];
} ADS;
```

User-defined or TINE system error

e.g. SetAlarm() with file_error :

All Alarms relying on the ‘generic’ alarm definitions land in alarm subsystem “SYSTEM” (5000) or “HARDWARE” (5001)

"File error”,
file_error,
0,
ALM_SYSTEM_SYSTEM
12
CF_TEXT,
64,
"file input/output error",
"",
"file/path name which caused the alarm“
“”

TINE Kernel 4.0.1

- Secondary Additions
  - Some new ‘error/status’ codes:
    - canbus_error (154)
    - profibus_error (155)
    - data_stale (156)
    - has_bitfield_tag (157)
    - has_structure_tag (157)

Analogous to sedac_error, gpib_error (generic alarm definitions)

Useful for ‘globals’ ?

Used internally to acquire bitfield/structure information

TINE Kernel 4.0.1

- Secondary Additions
  - Trap devices decorated with bitfield fields as per properties
    - e.g. Property “Status” delivers the whole bitfield for the chosen device
    - “Status.field1” just “field1” for the chosen device
    - Now: “Device1.field1” with property = “Status” will do the same !
    - Relevant to CDI devices with bitfields
  - Make a central logging entry contain up to 256 chars instead of 128

TINE Kernel 4.0.1

- Java:
  - All of the above (where java relevant) also implemented.
  - Bitfields tested.
  - Bug Fixes:
    - if looking for an export or local name entry in a fecid.csv file fails : don't take the last entry! (thanks Marcus)
    - new TLink().cancel() should not send anything ! ditto: TLink.execute() + .close(); (thanks Steve)
  - Still Missing: integration of the fec XML file reader (next week).

TINE Kernel 4.0.1

- Bug Fixes:
  - “Rare” alignment problem on Solaris fixed (thanks Olaf)
  - fixed inconsistency which allowed a link CANCEL request on a brand new contract to call the equipment function (thanks Steve)
  - fixed bug in CloseLink with a cdi hook that wasn't trapping the proper cdi link id. (thanks Honggong)

TINE Kernel 4.0.1

- Central Logger Class (ClsLog)
  - Public methods:
    - **public static Clog[]** getEntries(long start,long stop)
    - **public static Clog[]** getEntries(ClogFilter[] fltr)
    - **public static int** log(String text)
    - **public static int** log(String text,ClogPriority priority,ClogStatus status)
    - **public static int** log(String text,String Context,String tag,String caller, ClogPriority priority,ClogStatus status);
  - Public enumerations:

```
public static enum ClogStatus
{
  CLOG_STATUS_NONE,
  CLOG_STATUS_INFO,
  CLOG_STATUS_WARN,
  CLOG_STATUS_ERR
};
public static enum ClogPriority
{
  CLOG_PRIORITY_NONE,
  CLOG_PRIORITY_USEFUL,
  CLOG_PRIORITY_IMPORTANT,
  CLOG_PRIORITY_URGENT
}
```

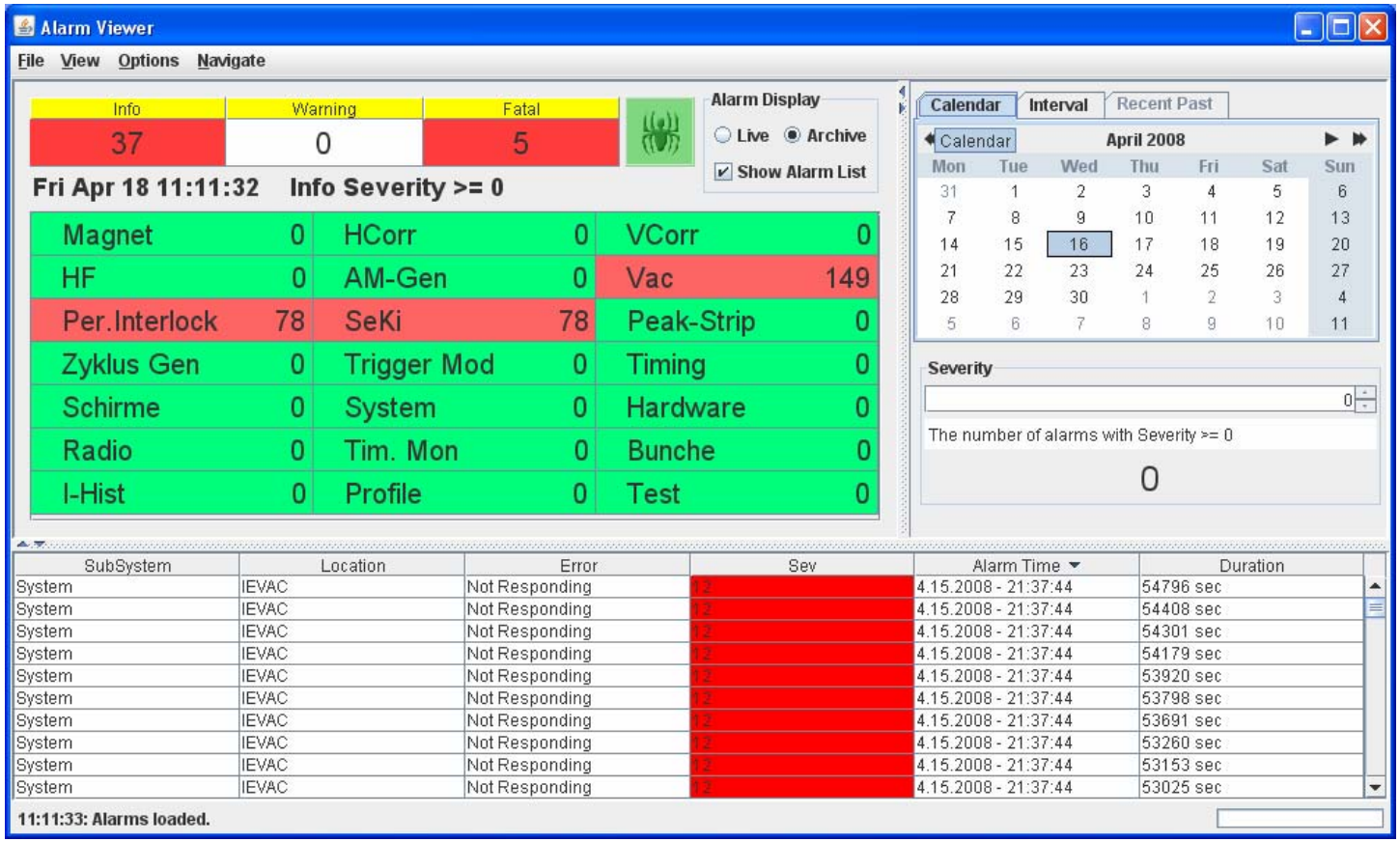
TINE Kernel 4.0.1

```
public static void main (String[] args)
{
  ClogFilter[] f = new ClogFilter[1];
  long stp = System.currentTimeMillis();
  long strt = stp - 3600000 * 24;
  f[0] = new ClogFilter(null,null,null,"karol",0,0,strt,stp);
  Clog[] clg = null;
  try
  { // just use the one filter; if an array of filter is passed they are ORed !
    clg = ClsLog.getEntries(f);
  }
  catch (IOException e1)
  {
    e1.printStackTrace();
  }
  if (clg != null) for (Clog c: clg) System.out.println( c.toString() );

  ClsLog.log("java test entry");

  ClsLog.log("java test entry with a bit more information","TEST","TEST",System.getProperty

  try { Thread.sleep(1000);}
  catch (InterruptedException e){}

  try
  {
    clg = ClsLog.getEntries(f);
  }
  catch (IOException e1)
  {
    e1.printStackTrace();
  }
  if (clg != null) for (Clog c: clg) System.out.println( c.toString() );
```

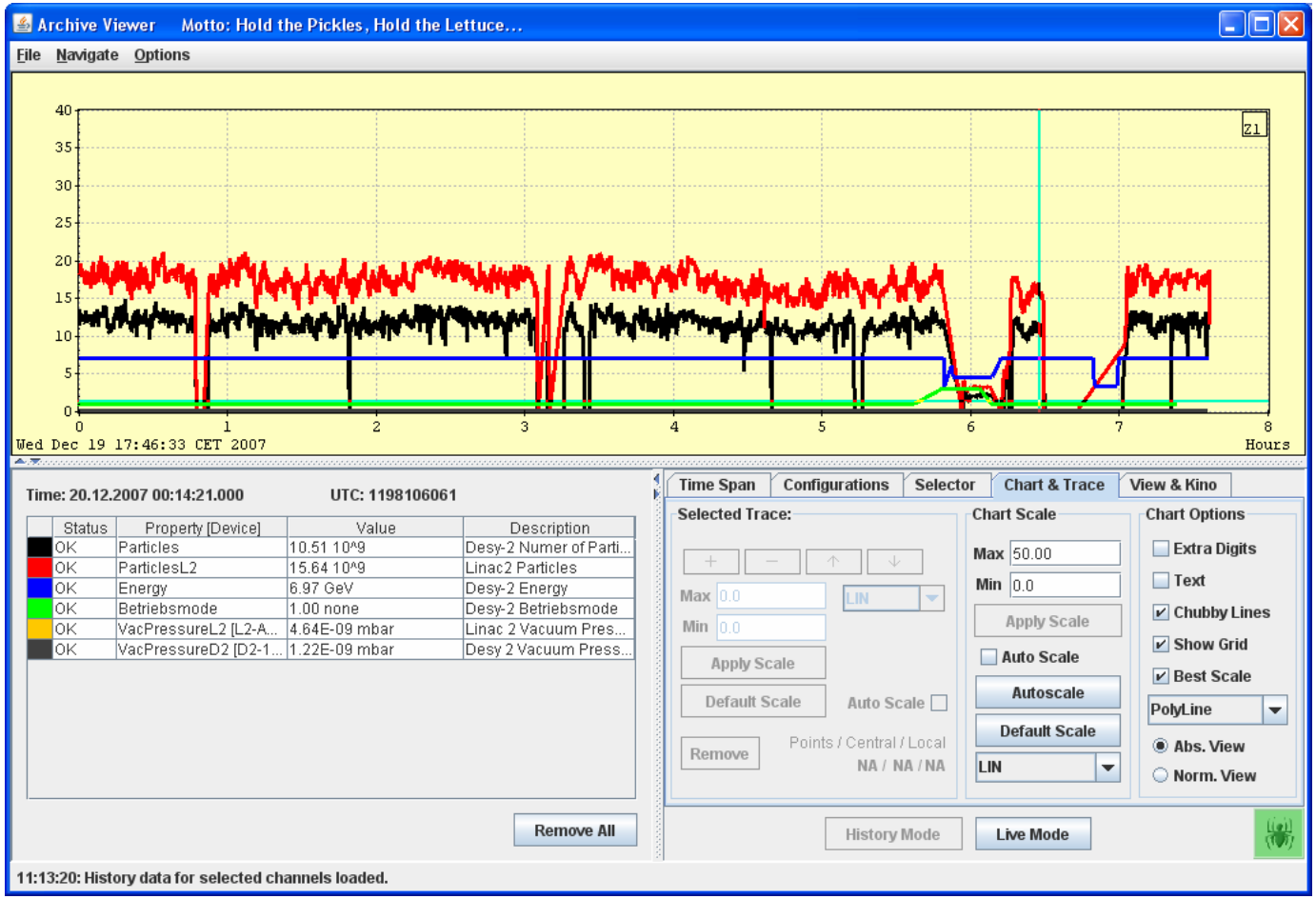
Central Services

- ENS
  - forwarding bug when run on Solaris fixed
- CLOG
  - logging entries can be 256 chars
- Globals
  - use FLTINT for the keyword data types ?
- Lots of FLASH Cryo Channels now available via epics2Tine (thanks Albert, Bernd)
- Mantis : now being used
  - English ?
  - Assign to ???

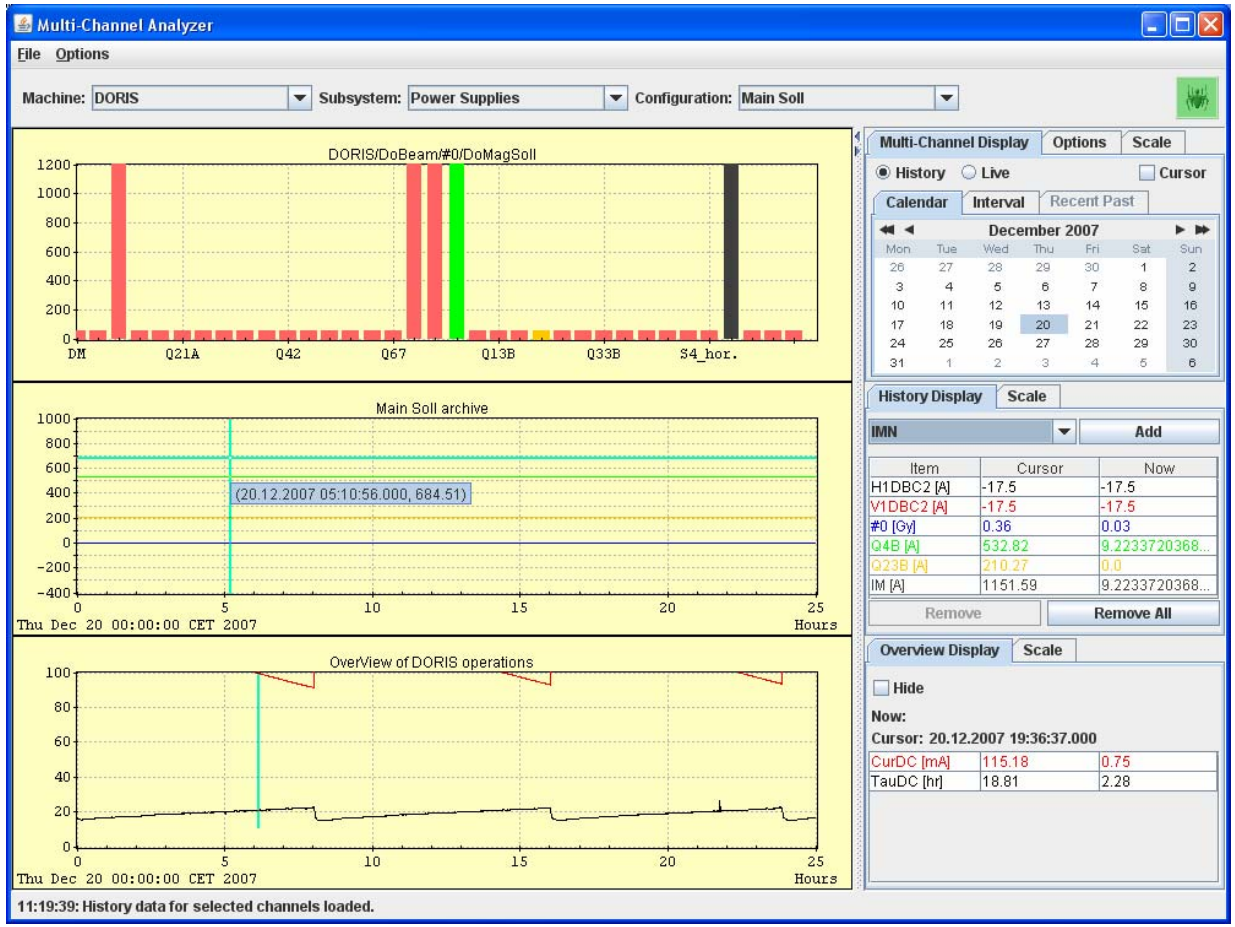
Java System Apps



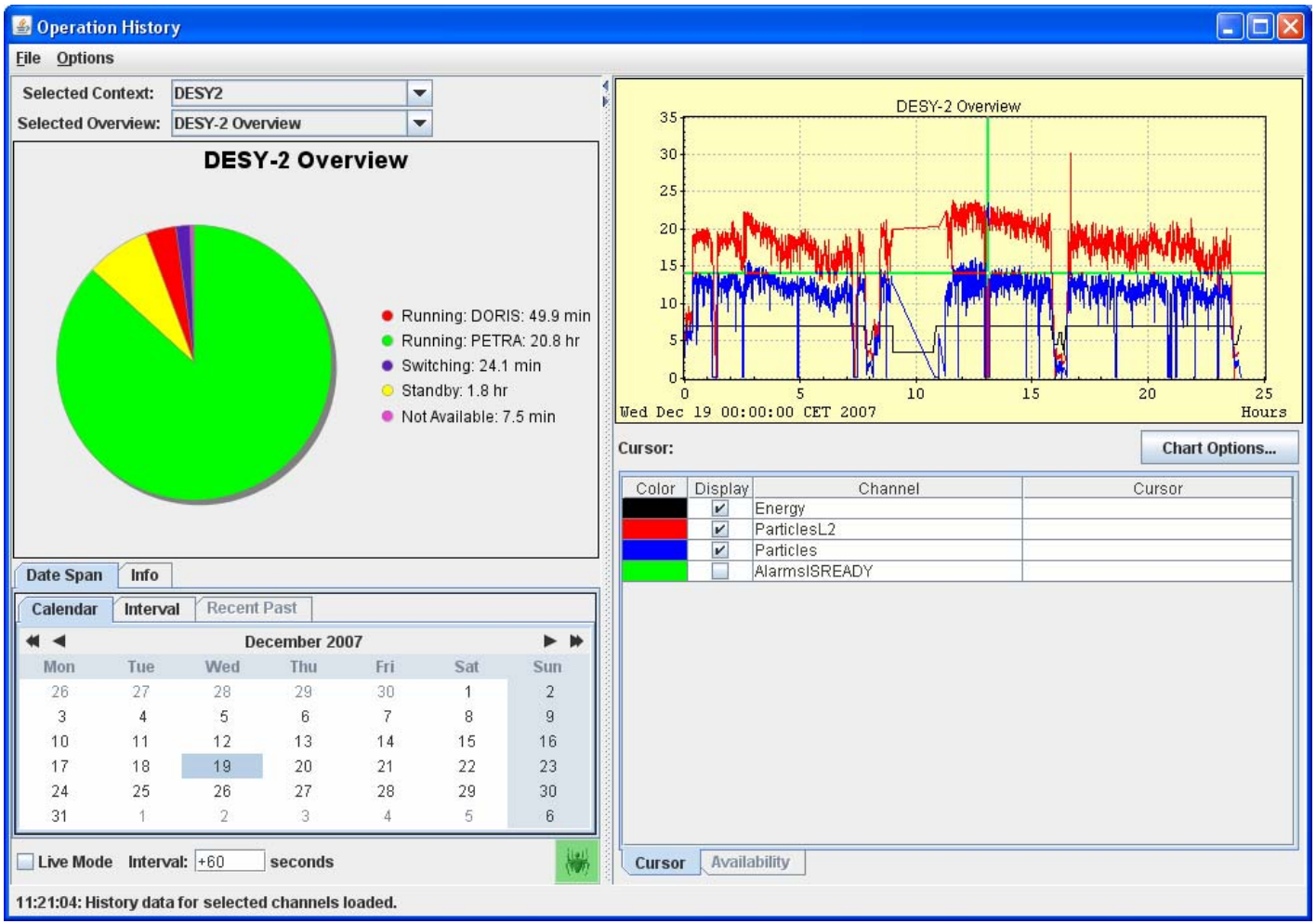
Java System Apps



Java System Apps



Java System Apps



Scope Trace Viewer